Reliable Fiducial Detection in Natural Scenes

David Claus and Andrew W. Fitzgibbon

Department of Engineering Science
University of Oxford, Oxford OX1 3BN
{dclaus,awf}@robots.ox.ac.uk

Abstract. Reliable detection of fiducial targets in real-world images is addressed in this paper. We show that even the best existing schemes are fragile when exposed to other than laboratory imaging conditions, and introduce an approach which delivers significant improvements in reliability at moderate computational cost. The key to these improvements is in the use of machine learning techniques, which have recently shown impressive results for the general object detection problem, for example in face detection. Although fiducial detection is an apparently simple special case, this paper shows why robustness to lighting, scale and foreshortening can be addressed within the machine learning framework with greater reliability than previous, more ad-hoc, fiducial detection schemes.

1 Introduction

Fiducial detection is an important problem in real-world vision systems. The task of identifying the position of a pre-defined target within a scene is central to augmented reality and many image registration tasks. It requires fast, accurate registration of unique landmarks under widely varying scene and lighting conditions. Numerous systems have been proposed which deal with various aspects of this task, but a system with reliable performance on a variety of scenes has not yet been reported.

Figure 1 illustrates the difficulties inherent in a real-world solution of this problem, including background clutter, motion blur [1], large differences in scale, foreshortening, and the significant lighting changes between indoors and out. These difficulties mean that a reliable general-purpose solution calls for a new approach. In fact, the paper shows how the power of machine learning techniques, for example as applied to the difficult problem of generic face detection [2], can benefit even the most basic of computer vision tasks.

One of the main challenges in fiducial detection is handling variations in scene lighting. Transitions from outdoors to indoors, backlit objects and in-camera lighting all cause global thresholding algorithms to fail, so present systems tend to use some sort of adaptive binarization to segment the features.

The problem addressed in this paper is to design a planar pattern which can be reliably detected in real world scenes. We first describe the problem, then cover existing solutions and present a new approach. We conclude by comparing the learning-based and traditional approaches.

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3024, pp. 469–480, 2004.

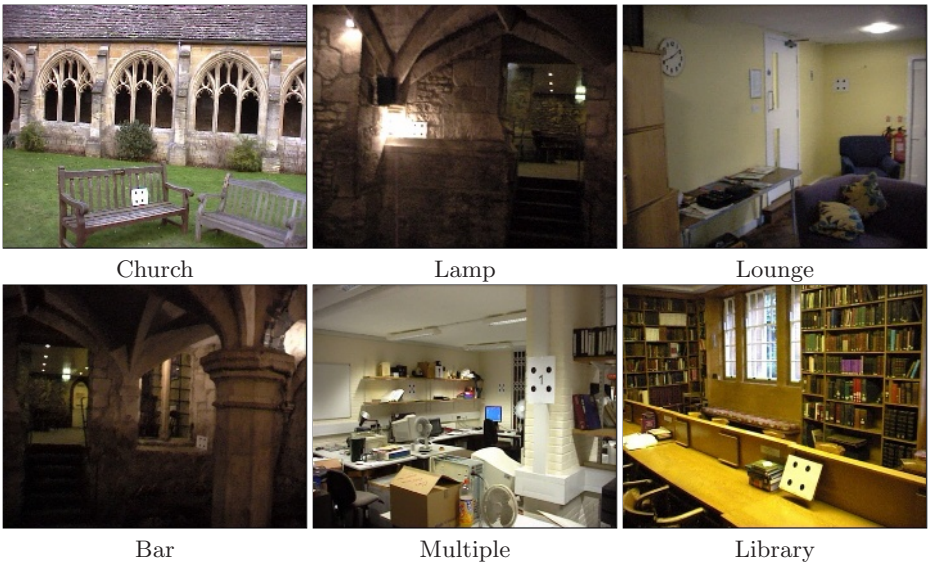


Fig. 1. Sample frames from test sequences. The task is to reliably detect the targets (four disks on a white background) which are visible in each image. It is a claim of this paper that, despite the apparent simplicty of this task, no technique currently in use is robust over a large range of scales, lighting and scene clutter. In real-world sequences, it is sometimes difficult even for humans to identify the target. We wish to detect the target with high reliability in such images.

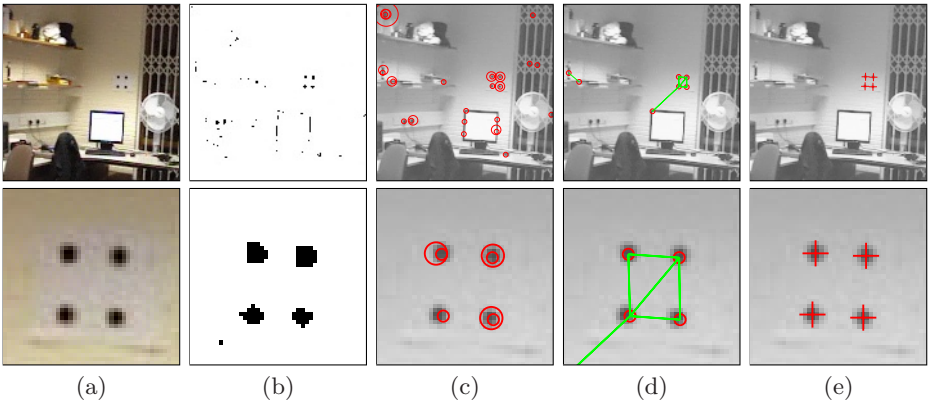


Fig. 2. Overall algorithm to locate fiducials. (a) Input image, (b) output from the fast classifier stage, (c) output from the full classifier superimposed on the original image. Every pixel has now been labelled as fiducial or non-fiducial. The size of the circles indicates the scale at which that fiducial was detected. (d) The target verification step rejects non-target fiducials through photometric and geometric checks. (e) Fiducial coordinates computed to subpixel accuracy.

2 Previous Work

Detection of known points within an image can be broken down into two phases: design of the fiducials, and the algorithm to detect them under scene variations. The many proposed fiducial designs include: active LEDs [3,4]; black and white concentric circles [5]; coloured concentric circles [6,7], one-dimensional line patterns [8]; squares containing either two-dimensional bar codes [9], more general characters [10] or a Discrete Cosine Transform [11]; and circular ring-codes [12, 13]. The accuracy of using circular fiducials is discussed in [14]. Three dimensional fiducials whose images directly encode the pose of the viewer have been propsed by [15]. We have selected a circular fiducial as the centroid is easily and efficiently measured to sub-pixel accuracy. Four known points are required to compute camera pose (a common use for fiducial detection) so we arrange four circles in a square pattern to form a target. The centre of the target may contain a barcode or other marker to allow different targets to be distinguished.

Naimark and Foxlin [13] identify non-uniform lighting conditions as a major obstacle to optical fiducial detection. They implement a modified form of homomorphic image processing in order to handle the widely varying contrast found in real-world images. This system is effective in low-light, in-camera lighting, and also strong side-lighting. Once a set of four ring-code fiducials have been located the system switches to tracking mode and only checks small windows around the known fiducials. The fiducial locations are predicted based on an inertial motion tracker.

TRIP [12] is a vision-only system that uses adaptive thresholding [16] to binarize the image, and then detects the concentric circle ring-codes by ellipse fitting. Although the entire frame is scanned on start-up and at specified intervals, an ellipse tracking algorithm is used on intermediate frames to achieve real-time performance. The target image can be detected 99% of the time up to a distance of 3 m and angle of 70 degrees from the target normal.

CyberCode [9] is an optical object tagging system that uses two-dimensional bar codes to identify object. The bar codes are located by a second moments search for guide bars amongst the regions of an adaptively thresholded [16] image. The lighting needs to be carefully controlled and the fiducial must occupy a significant portion of the video frame.

The AR Toolkit [10] contains a widely used fiducial detection system that tracks square borders surrounding unique characters. An input frame is thresholded and then each square searched for a pre-defined identification pattern. The global threshold constrains the allowable lighting conditions, and the operating range has been measured at 3 m for a 20×20 cm target [17].

Cho and Neumann [7] employ multi-scale concentric circles to increase their operating range. A set of 10 cm diameter coloured rings, arranged in a square target pattern similar to that used in this paper, can be detected up to 4.7 m from the camera.

Motion blur causes pure vision tracking algorithms to fail as the fiducials are no longer visible. Our learnt classifier can accomodate some degree of motion blur through the inclusion of relevant training data.

These existing systems all rely on transformations to produce invariance to some of the properties of real world scenes. However, lighting variation, scale changes and motion blur still affect performance. Rather than image preprocessing, we deal with these effects through machine learning.

2.1 Detection versus Tracking

In our system there is no prediction of the fiducial locations; the entire frame is processed every time. One way to increase the speed of fiducial detection is to only search the region located in the previous frame. This assumes that the target will only move a small amount between frames and causes the probability of tracking subsequent frames to depend on success in the current frame. As a result, the probability of successfully tracking through to the end of a sequence is the product of the frame probabilities, and rapidly falls below the usable range. An inertial measurement unit can provide a motion prediction [1], but there is still the risk that the target will fall outside the predicted region. This work will focus on the problem of detecting the target independently in each frame, without prior knowledge from the earlier frames.

3 Strategy

The fiducial detection strategy adopted in this paper is to collect a set of sample fiducial images under varying conditions, train a classifier on that set, and then classify a subwindow surrounding each pixel of every frame as either fiducial or not. There are a number of challenges, not least of which are speed and reliability.

We begin by collecting representative training samples in the form of 12×12 pixel images; larger fiducials are scaled down to fit. This training set is then used to classify subwindows as outlined in Figure 2. The classifier must be fast and reliable enough to perform half a million classifications per frame (one for the 12×12 subwindow at each location and scale) and still permit recognition of the target within the positive responses.

High efficiency is achieved through the use of a cascade of classifiers [2]. The first stage is a fast "ideal Bayes" lookup that compares the intensities of a pair of pixels directly with the distribution of positive and negative sample intensities for the same pair. If that stage returns positive then a more discriminating (and expensive) tuned nearest neighbour classifier is used. This yields the probability that a fiducial is present at every location within the frame; non-maxima suppression is used to isolate the peaks for subsequent verification.

The target verification is also done in two stages. The first checks that the background between fiducials is uniform and that the separating distance falls within the range for the scale at which the fiducials were identified. The second step is to check that the geometry is consitent with the corners of a square under perspective transformation. The final task is to compute the weighted centroid of each fiducial within the found target and report the coordinates.

The following section elaborates on this strategy; first we discuss the selection of training data, then each stage of the classification cascade is covered in detail.

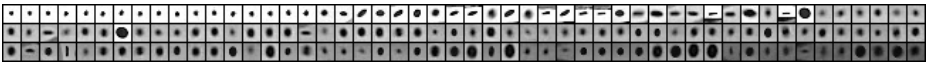


Fig. 3. Representative samples of positive target images. Note the wide variety of positive images that are all examples of a black dot on a white background.

3.1 Training Data

A subset of the positive training images is shown in Figure 3. These were acquired from a series of training videos using a simple tracking algorithm that was manually reset on failure. These samples indicate the large variations that occur in real-world scenes. The window size was set at 12×12 pixels, which limited the sample dot size to between 4 and 9 pixels in diameter; larger dots are scaled down by a factor of two until they fall within the specification. Samples were rotated and lightened or darkened to artificially increase the variation in the training set. This proves to be a more effective means of incorporating rotation and lighting invariance than *ad hoc* intensity normalization, as discussed in §5.

3.2 Cascading Classifier

The target location problem here is firmly cast as one of statistical pattern classification. The criteria for choosing a classifier are speed and reliability: the four subsampled scales of a 720×576 pixel video frame contain 522,216 subwindows requiring classification. Similar to [2], we have adopted a system of two cascading probes:

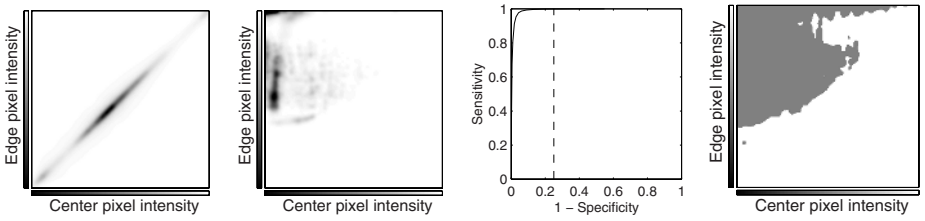
- fast Bayes decision rule classification on sets of two pixels from every window in the frame
- slower, more specific nearest neighbour classifier on the subset passed by the first stage

The first stage of the cascade must run very efficiently, have a near-zero false negative rate (so that any true positives are not rejected prematurely) and pass a minimal number of false positives. The second stage provides very high classification accuracy, but may incur a higher computational cost.

3.3 Cascade Stage One: Ideal Bayes

The first stage of the cascade constructs an ideal Bayes decision rule from the positive and negative training data distributions. These were measured from the training data and additional positive and negative images taken from the training videos. The sampling procedure selects two pixels from each subwindow: one at the centre of the dot and the other on the background. The distribution of the training data is shown in Figure 4.

The two distributions can be combined to yield a Bayes decision surface. If g_p and g_n represent the positive and negative distributions then the classification



(a) (b) (c) (d)

Fig. 4. Distribution of (a) negative pairs g_n and (b) positive pairs g_p used to construct the fast classifier. (c) ROC curve used to determine the value of α (indicated by the dashed line) which will produce the optimal decision surface given the costs of positive and negative errors. (d) The selected Bayes decision surface.

of a given intensity pair x is:

$$\text{classification}(x) = \begin{cases} +1 \text{ if } \alpha \cdot g_p(x) > g_n(x) \\ -1 \text{ otherwise} \end{cases} \tag{1}$$

where α is the relative cost of a false negative over a false positive. The parameter α was varied to produce the ROC curve shown in Figure 4c. A weighting of $\alpha = e^{12}$ produces the decision boundary shown in Figure 4d, and corresponds to a sensitivity of 0.9965 and a specificity of 0.75.

A subwindow is marked as a possible fiducial if a series of intensity pairs all lie within the positive decision region. Each pair contains the central point and one of seven outer pixels. The outer edge pixels were selected to minimize the number of false positives based on the above emiprical distributions.

The first stage of the cascade seeks dark points surrounded by lighter backgrounds, and thus functions is like a well-trained edge detector. Note however that the decision criteria is not simply $(edge - center) > threshold$ as would be the case if the center was merely required to be darker than the outer edge. Instead, the decision surface in Figure 4d encodes the fact that {dark center,dark edge} are more likely to be background, and {light center, light edge} are rare in the positive examples. Even at this early edge detection stage there are benefits from including learning in the algorithm.

3.4 Cascade Stage Two: Nearest Neighbour

Among the various methods of supervised statistical pattern recognition, the nearest neighbour rule [18] achieves consistently high performance [19]. The strategy is very simple: given a training set of examples from each class, a new sample is assigned the class of the nearest training example. In contrast with many other classifiers, this makes no *a priori* assumptions about the distributions from which the training examples are drawn, other than the notion that nearby points will tend to be of the same class.

For a binary classification problem given sets of positive and negative examples $\{p_i\}$ and $\{n_j\}$, subsets of $\mathbb{R}^d$ where d is the dimensionality of the input

vectors (144 for the image windows tested here). The NN classifier is then formally written as

$$\text{classification}(x) = -\text{sign}(\min_i \|p_i - x\|^2 - \min_j \|n_j - x\|^2)\,. \tag{2}$$

This is extended in the k-NN classifier, which reduces the effects of noisy training data by taking the k nearest points and assigning the class of the majority. The choice of k should be performed through cross-validation, though it is common to select k small and odd to break ties (typically 1, 3 or 5).

One of the chief drawbacks of the nearest neighbour classifier is that it is slow to execute. Testing an unknown sample requires computing the distance to each point in the training data; as the training set gets large this can be a very time consuming operation. A second disadvantage derives from one of the technique's advantages: that *a priori* knowledge cannot be included where it is available. We address both of these in this paper.

Speeding Up Nearest Neighbour. There are many techniques available for improving the performance and speed of a nearest neighbour classification [20]. One approach is to pre-sort the training sets in some way (such as *kd*-trees [21] or Voronoi cells [22]), however these become less effective as the dimensionality of the data increases. Another solution is to choose a subset of the training data such that classification by the 1-NN rule (using the subset) approximates the Bayes error rate [19]. This can result in significant speed improvements as k can now be limited to 1 and redundant data points have been removed from the training set. These data modification techniques can also improve the performance through removing points that cause mis-classifications.

We examined two of the many techniques for obtaining a training subset: condensed nearest neighbour [23] and edited nearest neighbour [24]. The condensed nearest neighbour algorithm is a simple pruning technique that begins with one example in the subset and recursively adds any examples that the subset misclassifies. Drawbacks to this technique include sensitivity to noise and no guarantee of the minimum consistent training set because the initial few patterns have a disproportionate affect on the outcome. Edited nearest neighbour is a reduction technique that removes an example if all of its neighbours are of a single class. This acts as a filter to remove isolated or noisy points and smooth the decision boundaries. Isolated points are generally considered to be noisy; however if no *a priori* knowledge of the data is assumed then the concept of noise is ill-defined and these points are equally likely to be valid. In our tests it was found that attempts to remove noisy points decreased the performance.

The condensing algorithm was used to reduce the size of the training data sets as it was desirable to retain "noisy" points. Manual selection of an initial sample was found to increase the generalization performance. The combined (test and training) data was condensed from 8506 positive and 19,052 negative examples to 37 positive and 345 negative examples.

Parameterization of Nearest Neighbour. Another enhancement to the nearest neighbour classifier involves favouring specific training data points through weighting [25]. In cases where the cost of a false positive is greater than the cost of a false negative it is desirable to weight all negative training data so that negative classification is favoured. This cost parameter allows a ROC curve to be constructed, which is used to tune the detector based on the relative costs of false positive and negative classifications.

We define the likelihood ratio to be the ratio of distances to the nearest negative and positive training examples:

$$likelihood\_ratio = nearest\_negative / nearest\_positive\,.$$

In the vicinity of a target dot there will be a number of responses where this ratio is high. Rather than returning all pixel locations above a certain threshold we locally suppress all non-maxima and return the point of maximum likelihood (similar to the technique used in Harris corner detection [26]; see [27] for additional details).

4 Implementation

Implementation of the cascading classifier described in the previous section is straightforward; this section describes the target verification step. Figure 2c shows a typical example of the classifier output, where the true positive responses are accompanied by a small number of false positives. Verification is merely used to identify the target amongst the positive classification responses; we outline one approach but there are any number of suitable techniques.

First we compute the Delaunay triangulation of all points to identify the lines connecting each positive classification with its neighbours. A weighted average adaptive thresholding of the pixels along each line identifies those with dark ends and light midsections. All other lines are removed; points that retain two or more connecting lines are passed to a geometric check. This check takes sets of four points, computes the transformation to map three of them onto the corners of a unit right triangle, and then applies that transformation to the remaining point. If the mapped point is close enough to the fourth corner of a unit square then retrieve the original grayscale image for each fiducial and return the set of weighted centroid target coordinates.

5 Discussion

The intention of this work was to produce a fiducial detector which offered extremely high reliability in real-world problems. To evaluate this algorithm, a number of video sequences were captured with a DV camcorder and manually marked up to provide ground truth data. The sequences were chosen to include the high variability of input data under which the algorithm is expected to be used. It is important also to compare performance to a traditional "engineered" detector, and one such was implemented as described in the appendix.

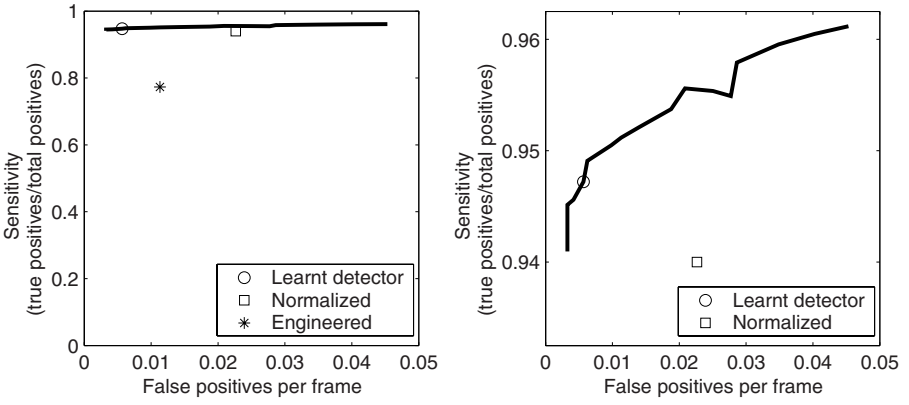


Fig. 5. (a) ROC curve for the overall fiducial detector. The vertical axis displays the percentage of ground truth targets that were detected. (b) An enlarged view of the portion of (a) corresponding to the typical operating range of the learnt detector. The drop in detection rate at 0.955 is an artefact of the target verification stage whereby some valid targets are rejected due to encroaching false positive fiducials.

Table 1. Success rate of target verification with various detectors. Normalizing each window prior to classification improves the success rate on some frames, but more false positive frames are introduced and the overall performance is worse. The engineered detector cannot achieve the same level of reliability as the learnt detector.

		Learnt detector		Normalized detector		Engineered detector	
Sequence	Targets	True	False	True	False	True	False
Church	300	98.3%	2.0%	99.3%	8.7%	46.3%	0.0%
Lamp	200	95.5%	0.5%	99.5%	3.0%	61.5%	0.0%
Lounge	400	98.8%	0.5%	96.5%	1.0%	96.5%	0.0%
Bar	975	89.3%	0.0%	91.3%	0.0%	65.7%	0.5%
Multiple	2100	95.2%	0.7%	93.5%	3.5%	83.0%	1.4%
Library	325	99.1%	0.6%	94.2%	0.0%	89.8%	5.2%
Summary	4300	94.7%	0.4%	94.0%	2.3%	77.3%	1.1%

The fiducial detection system was tested on six video sequences containing indoor/outdoor lighting, motion blur and oblique camera angles. The reader is encouraged to view the video of detection results available from [28].

Ground truth target coordinates were manually recorded for each frame and compared with the results of three different detection systems: learnt classifier, learnt classifier with subwindow normalization, and the engineered detector described in the appendix. Table 1 lists the detection and false positive rates for each sequence, while Table 2 lists the average number of positives found per frame. Overall, the fast classification stage returned just 0.33% of the subwindows as positive, allowing the classification system to process the average

720×576 frame at four scales in 120 ms. The operating range is up to 10 m with a 50 mm lens and angles up to 75 degrees from the target normal.

Normalizing each subwindow and classifying with normalized data was shown to increase the number of positives found. The target verification stage must then examine a larger number of features; since this portion of the system is currently implemented in Matlab alone it causes the entire algorithm to run slower. This is added to the increased complexity of computing the normalization of each subwindow prior to classification. By contrast, appending a normalized copy of the training data to the training set was found to increase the range of classification without significantly affecting the number of false positives or processing time. The success rate on the dimly lit bar sequence was increased from below 50% to 89.3% by including training samples normalized to approximate dim lighting.

Careful quantitative experiments comparing this system with the AR Toolkit (an example of a developed method of fiducial detection) have not yet been completed, however a qualitative analysis of several sequences containing both targets under a variety of scene conditions has been performed. Although the AR Toolkit performs well in the office, it fails under motion blur and when in-camera lighting disrupts the binarization. The template matching to identify a specific target does not incorporate any colour or intensity normalization and is therefore very sensitive to lighting changes. We deal with all of these variations through the inclusion of relevant training samples.

This paper has presented a fiducial detector which has superior performance to reported detectors. This is because of the use of machine learning. This detector demonstrated 95% overall performance through indoor and outdoor scenes including multiple scales, background clutter and motion blur. A cascade of classifiers permits high accuracy at low computational cost.

The primary conclusion of the paper is the observation that even "simple" vision tasks become challenging when high reliability under a wide range of operating conditions is required. Although a well engineered *ad hoc* detector can be tuned to handle a wide range of conditions, each new application and environment requires that the system be more or less re-engineered. In contrast,

Table 2. Average number of positive fiducial classifications per frame. The full classifier is only applied to the positive results of the fast classifier. This cascade allows the learnt detector to run faster and return fewer false positives than the engineered detector.

Sequence	True positives	Fast classifier	Full classifier	Normalized full classifier	Engineered detector
Church	4	5790	107	135	121
Lamp	4	560	23	30	220
Lounge	4	709	36	55	43
Bar	4	82	5	6	205
Multiple	7.3[†]	2327	79	107	96
Library	4	1297	34	49	82
Average	-	1794	47	64	128

[†]The Multiple sequence contains between 1 and 3 targets per frame.

with appropriate strategies for managing training set size, a detector based on learning can be retrained for new environments without significant architectural changes.

Further work will examine additional methods for reducing the computational load of the second classifier stage. This could include Locally Sensitive Hashing as a fast approximation to the nearest neighbour search, or a different classifier altogether such as a support vector machine.

References

1. Klein, G., Drummond, T.: Tightly integrated sensor fusion for robust visual tracking. In: Proc. BMVC. Volume 2. (2002) 787 –796
2. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. CVPR. (2001)
3. Neumann, U., Bajura, M.: Dynamic registration correction in augmented-reality systems. In: IEEE Virtual Reality Annual Int'l Symposium. (1995) 189–196
4. Welch, G., Bishop, G., et al.: The HiBall tracker: High-performance wide-area tracking for virtual and augmented environments. In: Proc. ACM VRST. (1999) 1–10
5. Mellor, J.P.: Enhanced reality visualization in a surgical environment. A.I. Technical Report 1544, MIT, Artificial Intelligence Laboratory (1995)
6. State, A., et al.: Superior augmented reality registration by integrating landmark tracking and magnetic tracking. In: SIGGRAPH. (1996) 429–438
7. Cho, Y., Neumann, U.: Multi-ring color fiducial systems for scalable fiducial tracking augmented reality. In: Proc. of IEEE VRAIS. (1998)
8. Scharstein, D., Briggs, A.: Real-time recognition of self-similar landmarks. Image and Vision Computing **19** (2001) 763–772
9. Rekimoto, J., Ayatsuka, Y.: Cybercode: Designing augmented reality environments with visual tags. In: Proceedings of DARE. (2000)
10. Kato, H., Billinghurst, M.: Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In: Int'l Workshop on AR. (1999) 85–94
11. Owen, C., Xiao, F., Middlin, P.: What is the best fiducial? In: The First IEEE International Augmented Reality Toolkit Workshop. (2002) 98–105
12. de Ipina, D.L., et al.: Trip: a low-cost vision-based location system for ubiquitous computing. Personal and Ubiquitous Computing **6** (2002) 206–219
13. Naimark, L., Foxlin, E.: Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In: ISMAR. (2002)
14. Efrat, A., Gotsman, C.: Subpixel image registration using circular fiducials. International Journal of Computational Geometry and Applications **4** (1994) 403–422
15. Bruckstein, A.M., Holt, R.J., Huang, T.S., Netravali, A.N.: New devices for 3D pose estimation: Mantis eyes, Agam paintings, sundials, and other space fiducials. International Journal of Computer Vision **39** (2000) 131–139
16. Wellner, P.: Adaptive thresholding for the digital desk. Technical Report EPC-1993-110, Xerox (1993)
17. Malbezin, P., Piekarski, W., Thomas, B.: Measuring artoolkit accuracy in long distance tracking experiments. In: 1st Int'l AR Toolkit Workshop. (2002)
18. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Information Theory **13** (1967) 57–67

19. Ripley, B.D.: Why do nearest-neighbour algorithms do so well? SIMCAT (Similarity and Categorization), Edinburgh (1997)
20. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. Machine Learning **38** (2000) 257–286
21. Sproull, R.F.: Refinements to nearest-neighbor searching in k-dimensional trees. Algorithmica **6** (1991) 579–589
22. Berchtold, S., Ertl, B., Keim, D.A., Kriegel, H.P., Seidl, T.: Fast nearest neighbor search in high-dimensional spaces. In: Proc. ICDE. (1998) 209–218
23. Hart, P.: The condensed nearest neighbor rule. IEEE Trans. Information Theory **14** (1968) 515–516
24. Duda, R., Hart, P., Stork, D.: Pattern Classification. 2nd edn. John-Wiley (2001)
25. Cost, S., Salzberg, S.: A weighted nearest neighbor algorithm for learning with symbolic features. Machine Learning **10** (1993) 57–78
26. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proceedings of the 4th ALVEY Vision Conference. (1988) 147–151
27. Claus, D.: Video-based surveying for large outdoor environments. First Year Report, University of Oxford (2004)
28. http://www.robots.ox.ac.uk/~dclaus/eccv/fiddetect.mpg.

Appendix: Engineered Detector

One important comparison for this work is how well it compares with traditional *ad hoc* approaches to fiducial detection. In this section we outline a local implementation of such a system.

Each frame is converted to grayscale, binarized using adaptive thresholding as described in [16], and connected components used to identify continuous regions. The regions are split into scale bins based on area, and under or over-sized regions removed. Regions are then rejected if the ratio of the covex hull area and actual area is too low (region not entirely filled or boundary is not continually convex), or if they are too eccentric (if the axes ratio of an ellipse with the same second moments is too high).